

TITLE OF THE INVENTION  
IMAGE PROCESSING METHOD AND APPARATUS, IMAGE PROCESSING  
SYSTEM, AND STORAGE MEDIUM

5 BACKGROUND OF THE INVENTION

This invention relates to an image processing method, apparatus, and system for modifying an original image, and a storage medium.

10 As digital cameras, photoscanners, and the like have prevailed in recent years, a photo image can be easily converted into digital data. Also, since output apparatuses represented by ink-jet printers have attained higher image quality and lower prices, the end  
15 user can easily output a photo onto a recording sheet at home.

Hence, with an apparatus such as a personal computer or the like that allows image processing, the end user edits an image sensed by, e.g., a digital  
20 camera as he or she so desires, and outputs the edit result via an ink-jet printer.

A process called posterization has been proposed as Japanese Patent Laid-Open No. 1-314389, and an illustration edit process has been proposed as Japanese  
25 Patent Laid-Open Nos. 3-91088 and 3-91087.

As the personal computer and printer that can implement an edit process of a digital photo image have higher performance and lower prices, for example, the end user can easily create name cards having his or her own face photo superimposed on them.

Also, a machine that can print photo stickers of a sensed image at that place has prevailed.

A case will be examined below wherein a photo is superimposed on a name card. Upon creating formal name cards, it is a common practice to superimpose an expressionless face photo. However, as such name cards sometimes produce stiffness, informal name cards are often required. In such case, the need for superimposing not only a face photo that the user desired but a more personalized photo created by editing the photo itself has arisen. As an edit process that meets such need, a monochrome edit process represented by sepia-tone conversion, an edit process for converting a photo into an illustration, and the like are known.

In this manner, various needs for photo images have increased recently.

However, in the conventional image processing apparatus, in order to edit a photo image to obtain an illustration-style image, the edges of the photo image are extracted and individual portions are painted in the same colors as those in the original image. Furthermore,

the user must finely set various parameters required for edge extraction, painting, and the like.

Hence, in order to obtain an illustration-style image, room for improvement still remains in terms of image quality. Also, not only complicated setups but also considerable user skills are required to obtain appropriate setups. Hence, it is difficult for the end user to attain such setups.

10

#### SUMMARY OF THE INVENTION

Accordingly, it is an object of the present invention to provide an image processing method, apparatus, and system which can easily perform an effective simulate illustration process for image data, and a storage medium.

According the present invention, the foregoing object is attained by providing an image processing method comprising: the modify step of modifying a first image to obtain a second image on the basis of a first signal obtained by extracting an edge of the first image, and a second signal obtained by reducing the number of tone levels.

It is another object of the present invention to provide an image processing method, apparatus, and

system which allows the user to easily set up details of simulate illustration process, and a storage medium.

According the present invention, the foregoing object is attained by providing an image processing  
5 method further comprising: the instruction input step of inputting a user instruction that selects a desired one of a plurality of modify modes; and the image process step of executing an image process for image data using the modify mode corresponding to the user instruction,  
10 and wherein the plurality of modify modes include an illustration mode for converting the first image into the second image in the modify step.

It is another object of the present invention to provide an image processing method, apparatus, and  
15 system which can easily perform an effective simulate illustration process even for rasterized image data, and a storage medium.

According the present invention, the foregoing object is attained by providing an image processing  
20 method further comprising: the segmentation step of segmenting the first image into a plurality of regions; and the setting step of setting processing contents of each of the segmented regions, and wherein the modify step includes the step of modifying the first image in  
25 units of regions to obtain the second image.

The invention is particularly advantageous since a simulate illustration process can be done for image data more effectively in brighter tone than a conventional method.

5        Also, the user can easily set up details of simulate illustration process.

Furthermore, a recording medium on which an image that has undergone a novel process closely resembling a hand illustration is formed can be provided.

10        Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures thereof.

15

#### BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of the specification,  
20        illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is a block diagram showing the arrangement of an image processing system according to an embodiment  
25        of the present invention;

Fig. 2 is a block diagram of a printer driver;

Fig. 3 is a flow chart showing an image process in the printer driver;

Fig. 4 is a flow chart showing a histogram generation process;

5 Fig. 5 shows an example of a normal filter;

Figs. 6A to 6C show other examples of filters (5 × 5);

Fig. 7 shows still another example of a filter (3 × 3);

10 Fig. 8 is a flow chart showing a simulate illustration process;

Fig. 9 is a flow chart showing an image edit process according to the second embodiment of the present invention;

15 Fig. 10 shows an example of a selection window for an image to be edited;

Fig. 11 shows an example of a selection window for a modify mode;

20 Fig. 12 shows an example of a setup window for details of an illustration mode;

Fig. 13 is a flow chart showing an original image data generation process;

Fig. 14 is a flow chart showing a simulate illustration process;

25 Fig. 15 shows an example of a filter process for an image edge upon batch-processing an image;

Figs. 16A and 16B show an example of area segmentation upon obliquely setting an image;

Fig. 17 is a flow chart showing a simulate illustration process according to the third embodiment  
5 of the present invention;

Figs. 18A and 18B show examples of special filters in the third embodiment;

Figs. 19A to 19C show a user interface used to set up an image process according to the fourth embodiment  
10 of the present invention;

Fig. 20A shows an example of an original image (JPEG);

Fig. 20B shows an example of an illustration edit result of the original image shown in Fig. 20A using a 3  
15  $\times$  3 filter;

Fig. 20C shows an example of an illustration edit result of the original image shown in Fig. 20A using a 5  
 $\times$  5 filter;

Fig. 20D shows an example of an illustration edit  
20 & monochrome edit result of the original image shown in Fig. 20A;

Fig. 21A shows an example of an original image (TIFF);

Fig. 21B shows an example of an illustration edit  
25 result of the original image shown in Fig. 21A using a 3  
 $\times$  3 filter; and

Fig. 21C shows an example of an illustration edit result of the original image shown in Fig. 21A using a 5 × 5 filter.

5 DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention will now be described in detail in accordance with the accompanying drawings.

10 <First Embodiment>

•System Arrangement

Fig. 1 shows an example of an image processing system according to this embodiment. A printer 105 such as an ink-jet printer or the like, and a monitor 106 are  
15 connected to a host computer 100.

The host computer 100 has as software application software programs 101 such as a wordprocessor, spreadsheet, Internet browser, and the like, an OS (Operating System) 102, a printer driver 103 for  
20 processing various drawing commands (image drawing command, text drawing command, graphics drawing command) indicating output images, which are issued by the application software programs 101 to the OS 102, and generating print data, and a monitor driver 104 for  
25 processing various drawing commands issued by the



application software programs 101 and displaying data on the monitor 106.

Reference numeral 112 denotes an instruction input device; and 113, its device driver. For example, a mouse  
5 that points various kinds of information displayed on the monitor 106 to issue various instructions to the OS 102 is connected. Note that other pointing devices such as a trackball, pen, touch panel, and the like, or a keyboard may be connected in place of the mouse.

10 The host computer 100 comprises, as various kinds of hardware that can run these software programs, a central processing unit (CPU) 108, hard disk (HD) 107, random-access memory (RAM) 109, read-only memory (ROM) 110, and the like.

15 As an example of the image processing system shown in Fig. 1, Windows98 available from Microsoft Corp. is installed as an OS in a popular PC-AT compatible personal computer available from IBM Corp., desired application programs that can implement printing are  
20 installed, and a monitor and printer are connected to the personal computer.

In the host computer 100, each application software program 101 generates output image data using text data which is classified into text such as  
25 characters or the like, graphics data which is classified into graphics such as figures or the like,

image data which is classified into a natural image or the like, and so forth. Upon printing out the output image data, the application software program 101 sends a print-out request to the OS 102. At this time, the  
5 application software program 101 issues a drawing command group which includes a graphics drawing command corresponding to graphics data, and an image drawing command corresponding to image data to the OS 102.

Upon receiving the output request from the  
10 application software program 101, the OS 102 issues a drawing command group to the printer driver 103 corresponding to an output printer. The printer driver 103 processes the print request and drawing command group input from the OS 102, generates print data that  
15 the printer 105 can print, and transfers the print data to the printer 105. When the printer 105 is a raster printer, the printer driver 103 performs an image correction process for the drawing commands from the OS 102, and then rasterizes the commands in turn on the RGB  
20 24-bit page memory. Upon completion of rasterization of all the drawing command, the printer driver 103 converts the contents of the RGB 24-bit page memory into a data format that the printer 105 can print, i.e., CMYK data, and transfers the converted data to the printer 105.

25 Note that the host computer 100 can connect a digital camera 111 which senses an object image and

generates RGB image data, and can load and store the sensed image data in the HD 107. Note that the image data sensed by the digital camera 111 is encoded by JPEG. The sensed image data can be transferred as image data  
5 to the printer 105 after it is decoded by the printer driver 103.

•Printer Driver Process

The process executed by the printer driver 103 will be explained below with reference to Fig. 2.

10 In the printer driver 103, an image correction processing unit 120 executes an image correction process and image edit process (to modify image data; to be described later) for color information of an image drawing command included in the drawing command group  
15 received from the OS 102. Then, a printer correction processing unit 121 rasterizes the drawing commands in accordance with the processed color information to generate dot image data on the RGB 24-bit page memory. The unit 121 executes a masking process, gamma  
20 correction process, quantization process, and the like in units of pixels in correspondence with color reproducibility of the printer to generate CMYK data depending on the printer characteristics, and transfers them to the printer 105.

25 An image process for an original image indicated by an image drawing command in the image correction

processing unit 120 will be explained below with  
reference to the flow chart in Fig. 3. Note that an  
original image indicated by a graphics drawing command  
or text drawing command does not undergo a process to be  
5 described below. Assume that an original image is stored  
in a predetermined area in the RAM 109.

The image correction processing unit 120 of this  
embodiment decodes the original image if it is encoded  
by, e.g., JPEG (S20, S21), as shown in Fig. 3. After  
10 that, the unit 120 executes a histogram generation  
process (S22) and an image correction process (S23) in  
accordance with the generated histogram, and then  
executes a simulate illustration process (S24) in  
accordance with the generated histogram.

15 The image that has undergone the simulate  
illustration process is output to the printer 105 as  
printable data via the printer correction processing  
unit 121, and is printed out onto a storage medium.

•Generation of Luminance Histogram

20 Fig. 4 is a flow chart showing the histogram  
generation process in step S22.

Referring to Fig. 4, when the control enters an  
original image luminance histogram generation routine in  
step S1, the selection ratio of pixels used to generate  
25 a luminance histogram to those of an original image is  
determined in step S2. In this embodiment, when image

data to be processed has 350,000 pixels, a luminance histogram is generated for all the pixels (selection ratio = 1 (or 100%)). When image data having more than 350,000 pixels is input, pixel selection (sampling) is done in accordance with the ratio of the total number of pixels to 350,000 pixels. For example, when image data having 3,500,000 pixels is input, the selection ratio is  $3,500,000/350,000 = 10$ , and a luminance histogram is generated at a ratio of one pixel per 10 pixels (selection ratio = 10 (or 10%)). The selection ratio  $n$  in this embodiment is given by:

$$n = \text{int}(\text{total number of pixels of image data to be processed} / 350,000 \text{ as reference number of pixels})$$

(for  $n = 1$  when  $n < 1$ ,  $n$  is a positive number)

Subsequently, a counter that manages a line number is reset or set at a predetermined initial value in step S3, and that counter is incremented to indicate the line number of the line of interest in step S4.

In this embodiment, since pixel thinning (sampling) is done in units of lines, when the selection ratio =  $n$ , and the remainder obtained upon dividing the line number by  $n$  is 0, pixels which belong to that line are selected as those to be processed (S5 = YES). For example, when the selection ratio = 10, and the remainder obtained upon dividing the line number by 10

is 0, pixels which belong to that line are selected as those to be processed.

If the line of interest is a line to be thinned out, i.e., a line which is not to be processed, the flow returns to step S4. If the line of interest is a line to be processed, the flow advances to step S6, each of pixels which belong to the line of interest is selected in turn, and luminance conversion and chromaticity conversion are done for that pixel of interest.

10 Luminance conversion and chromaticity conversion in this embodiment are attained by:

$$Y \text{ (luminance)} = \text{int}(0.30R + 0.59G + 0.11B)$$

(Y is a positive integer)

$$C1 \text{ (chromaticity)} = R - Y$$

15  $C2 \text{ (chromaticity)} = B - Y$

Note that luminance and chromaticity conversion equations are not limited to the above ones, but may use various other equations.

In this embodiment, in order to improve the detection precision of highlight and shadow points, a saturation S of the pixel of interest is calculated by:

$$\text{Saturation } S = \sqrt{(C1^2 + C2^2)}$$

It is then checked if the saturation S is larger than a predetermined saturation value (Sconst) (S7). If S > Sconst, information of that pixel is not reflected in the luminance histogram.

That is, if  $S > S_{const}$ , the flow returns to step S6 not to reflect data of that pixel in the subsequent processes. In general, since the saturation of the highlight point is given by the average saturation of a high-luminance pixel group, that saturation value includes errors produced by color cast. Hence, a pixel which has a saturation higher than the predetermined value is preferably excluded from a highlight point calculation.

10        An example of the effect of this process will be explained below. For example, the luminance  $Y$  and saturation  $S$  of a yellow pixel ( $R = G = 255$ ,  $B = 0$ ) are respectively 226 and 227 from the above equations. That is, this pixel has a very high luminance and assumes a  
15        sufficiently high saturation color. It is appropriate to determine that such pixel is an original yellow pixel rather than to determine that the pixel is obtained as a result of yellow cast of an achromatic pixel. When a luminance histogram including such  
20        high-luminance/high-saturation pixels is generated, errors are produced in the highlight point detection result. Hence, in this embodiment, the predetermined saturation ( $S_{const}$ ) is determined, and pixels having saturation beyond this value are excluded from the  
25        luminance histogram. In this manner, the highlight point detection result can be prevented from suffering any

errors due to such high-saturation pixels, and the detection precision of the highlight point can be improved.

However, if the average density of an image need  
5 only be detected, as will be described later, the aforementioned process in step S7 in this embodiment may be omitted.

After the decision in step S7, a luminance histogram is generated using pixels that satisfy a  
10 condition ( $S \leq S_{const}$ ) (S8). Since pixel data RGB to be processed in this embodiment are 8-bit (256 tone levels) data, the luminance Y is also converted into a depth of 256. Hence, the luminance histogram is obtained by counting the frequency of occurrence of pixels  
15 corresponding to each of 256-level luminance values ranging from 0 to 255.

Note that the calculated chromaticity values C1 and C2 can be used as data for calculating the average chromaticity of pixels having individual luminance  
20 values in, e.g., a color cast correction process. Hence, in this embodiment, data are held as follows. Three members, i.e., the frequency of occurrence, C1 accumulated value, and C2 accumulated value, are set in the format of a structure array variable which has an  
25 index range from 0 to 255, and the computation result of



each pixel is reflected in each member having the luminance value of that pixel as an index.

Upon completion of processing for the pixel of interest, it is checked if all the pixels in the line of interest have been processed (S9). If pixels to be processed still remain in the line of interest, the flow returns to step S6 to repeat the processes in step S6 and the subsequent steps. Upon completion of processing for all the pixels in the line of interest, it is checked in step S10 if lines to be processed still remain. If processing for all the lines is complete, the flow ends in step S11; otherwise, the flow returns to step S4 to select the next line as the line of interest, thus repeating the aforementioned processes.

In this manner, since the luminance histogram is generated while selecting pixels of original image data, it can be generated using a minimum required number of pixels in consideration of improvement of the precision upon detecting highlight and shadow points.

#### •Image Correction Process

In step S23 in Fig. 3, the original image undergoes an image correction process based on the luminance histogram obtained in step S22. For example, highlight and shadow points of the original image are detected based on the luminance histogram, and image correction processes are made based on the obtained

highlight and shadow points. More specifically, in the color correction processes, color cast correction for correcting color cast of the original image, exposure correction for correcting the contrast of luminance to  
5 optimize the exposure state of the original image, saturation correction for improving the appearance of colors of an output image, and the like are done. Note that these image correction processes can use known methods, and a detailed description thereof will be  
10 omitted. Note that the correction processes described in step S23 may be omitted in this invention.

•Simulate illustration Process

In step S24 in Fig. 3, the original image corrected in step S23 undergoes a simulate illustration  
15 process based on the luminance histogram obtained in step S22. This embodiment is characterized in that the original image sensed by, e.g., the digital camera 111 undergoes the simulate illustration process for converting the original image into a hand-illustrated  
20 like image.

The principle of the simulate illustration process in this embodiment will be explained first. In this embodiment, the original image is filtered using a 5 pixels  $\times$  5 pixels (to be simply referred to as 5  $\times$  5  
25 hereinafter) filter that satisfies a predetermined condition with respect to the original image to be

converted into an image in which edges are extracted  
(emphasized) and color tone is saved. Fig. 5 shows an  
example of this filter. For example, when an original  
image shown in Fig. 20A is filtered using a filter 40  
5 shown in Fig. 5, an image shown in Fig. 20C is obtained.  
As can be seen from Fig. 20C, since the edges of the  
original image as a photo image are extracted and the  
number of tone levels decreases and the brightness of  
the image increases, the original image is converted  
10 into an image that gives an impression closely  
resembling a hand illustration. Note that the image  
shown in Fig. 20A is sensed by PowerShotA5 (tradename)  
as a digital camera available from CANON Corp., and has  
810,000 pixels.

15 A filter that implements the simulate illustration  
process of this embodiment will be explained below.

As a filter for extracting edges of an image, for  
example, a Laplacian filter is known. In general, in the  
Laplacian filter, the coefficient (weight) of the pixel  
20 of interest located at the center of the filter is set  
to be larger than those of surrounding pixels to allow  
extraction of a density change point in the image, i.e.,  
an edge. Normally, the sum total of coefficients in the  
Laplacian filter for edge extraction is "0".

25 The filter for the simulate illustration process  
(to be simply referred to as a "filter" hereinafter) in

this embodiment preferably uses, e.g., the filter 40 shown in Fig. 5, as described above, i.e., a  $5 \times 5$  filter consisting of coefficients shown in Fig. 5. In the filter 40, since the coefficient of the pixel of interest is set at "26", and coefficients of all surrounding pixels are set at "-1", the sum total of coefficients is "2".

The filter in this embodiment is not limited to the filter 40 shown in Fig. 5, and the sum total of coefficients need only be equal to or larger than zero. Figs. 6A to 6C show other examples of the filter. Fig. 6A shows a filter 61 in which the coefficient of the pixel of interest is set at "25" by subtracting "1" from that of the filter 40, and Fig. 6B shows a filter 62 in which the coefficient of the pixel of interest is set at "27" by adding "1" to that of the filter 40. An image filtered using the filter 61 becomes darker than that obtained by the filter 40, and an image filtered using the filter 62 becomes brighter than that obtained by the filter 40. Hence, according to the characteristics of these filtering results, if the filter 40 is used as a normal filter, the filters 61 and 62 are respectively used as relatively dark and bright filters. Such filter characteristics can be easily understood from the following fact. That is, assuming that a uniform density area of an image is filtered, if

the sum total of filter coefficients is "2", the signal value, i.e., its brightness is doubled; if the sum total of filter coefficients is "3", its brightness is tripled.

Fig. 6C shows a filter 63 in which coefficients of  
5 pixels around the pixel of interest in the filter 40 are  
thinned out. In this filter, the coefficient of the  
pixel of interest is set at "18" to maintain the sum  
total of coefficients to be "2". According to the  
filtering process using the filter 63, since the number  
10 of coefficients decreases, the computation volume upon  
filtering can be reduced, and the processing speed can  
be improved. Note that the coefficients in the filter 62  
are thinned out in consideration of edge detection in  
the vertical, horizontal, and oblique directions.

15 The  $5 \times 5$  filters have been explained as examples  
of the filters used in this embodiment. This embodiment  
can also be practiced using filters having other sizes.  
For example, the filter size can be defined by  $n \times n$  ( $n$   
 $= 2 \times m + 1$  ( $m$  is an integer equal to or larger than 1)).  
20 Also, the filter may have different vertical and  
horizontal sizes. Fig. 7 shows a  $3 \times 3$  filter as an  
example of the filter having another size. When an  
original image shown in Fig. 20A is filtered using the  $3$   
 $\times 3$  filter shown in Fig. 7, an image shown in Fig. 20B  
25 is obtained. In the image shown in Fig. 20B as well,  
edges are extracted from the original image as a photo

image and the number of tone levels decreases, and the brightness increases. However, the illustration edit effect is lower than that of the image shown in Fig. 20C based on the  $5 \times 5$  filter 40. This is because the  $5 \times 5$  filter acts to express the extracted edges thicker.

In the simulate illustration process in this embodiment, the thickness of the extracted edges in the converted image changes depending on the filter size. Hence, an optimal filter size upon simulate illustration process is not limited to  $5 \times 5$  or  $3 \times 3$  described in this embodiment, and depends on the image size or resolution of an original image, fineness of an object in an image, and the like. For example, when a large filter is applied to a large-size or high-resolution image or an image including a coarse object, or when a small filter is applied to a small-size or low-resolution image or an image including a fine object, the extracted edges become thicker in the former case or thinner in the latter case. On the other hand, the filter size may be set in correspondence with the brightness of an object. Furthermore, the filter size may be set in correspondence with the processing speed required.

In the simulate illustration process in this embodiment, the number of tone levels of image data is

decreased, but the brightness need not always be increased.

Such filter process may be automatically selected in correspondence with the size or resolution of an  
5 image or the fineness of an object, or may be selected in accordance with a manual instruction of the user. Hence, the present invention includes provision of a user interface for manual selection.

In step S24 in Fig. 3, the original image  
10 corrected in step S23 undergoes the simulate illustration process using the aforementioned filter. Fig. 8 is a flow chart showing the simulate illustration process in this embodiment. In step S81, the brightness of an image is detected on the basis of the luminance  
15 histogram generated in step S22 above. As the detection method, the average value, median, or maximum frequency value may be calculated based on the luminance histogram, and is compared with a predetermined threshold value to detect a plurality of brightness levels. For example, by  
20 setting a predetermined range corresponding to a "normal" brightness level in advance, an image which has brightness higher than that predetermined range is determined to have "bright" level, and an image which has brightness lower than the predetermined range is  
25 determined to have "dark" level, thus allowing detection of three different brightness levels.

Based on the brightness level detected in step S81,  
a filter is set in step S82, S83, or S84. That is, if  
the original image has "bright" level, the filter 61 as  
a relatively dark filter is set in step S82 to decrease  
5 the brightness of the conversion result; if the original  
image has "normal" level, the filter 40 as a normal  
filter is set in step S83; or if the original image has  
"dark" level, the filter 62 as a relatively bright  
filter is set in step S84 to increase the brightness of  
10 the converted image.

In step S85, the original image is filtered using  
the selected filter, thus implementing appropriate  
simulate illustration process.

Note that the number of brightness levels to be  
15 determined is not particularly limited, and can be set  
in correspondence with filters that can provide  
appropriate effects in units of levels.

In this embodiment, since the printer driver 105  
executes the simulate illustration process, the entire  
20 region of the original image cannot undergo a batch  
process. For this reason, the original image is  
segmented into region blocks each consisting of a  
plurality of lines, and the illustration process is done  
in units of blocks. Hence, in order to avoid block  
25 boundaries from becoming conspicuous in the converted  
image, the filter size is changed to a smaller one, e.g.,



3 x 5 or the like near the final line of each block. In this manner, continuity of block boundaries can be maintained in the converted image.

As described above, according to this embodiment,  
5 an original image can easily undergo simulate  
illustration process only with modifying image data, and  
a hand-illustrated like image can be obtained with an  
atmosphere of the original image, since its lightness  
increases. As the illustration-edited image has a  
10 smaller number of tone levels than the original image,  
the data size of the image can be reduced.

In this embodiment, the luminance histogram of an  
original image is generated, and the original image  
undergoes the image correction process based on that  
15 luminance histogram. After that the simulate  
illustration process is executed based on the luminance  
histogram. However, the image correction process based  
on the luminance histogram need not always be required.  
When the simulate illustration process alone is done  
20 immediately after the luminance histogram is generated,  
the processing speed can be improved.

When the user manually designates brightness he or  
she wants in correspondence with the exposure state or  
the like of an original image, the histogram generation  
25 process in step S22 may be omitted.

On the other hand, filter control in the simulate illustration process may be done on the basis of parameters based on the image correction process such as contrast information between the background and object, scene information to which an image belongs, and the like, which are used as discrimination conditions upon executing color correction or the like in the image correction process.

In this embodiment, a histogram of a luminance signal Y is generated. Alternatively, brightness detection may be done based on the histogram of G components without executing any luminance/chromaticity conversion of original image data. In this case, an increase in processing speed can be expected since no luminance/chromaticity conversion is done.

In this embodiment, an original image is RGB data. Of course, the present invention can be applied even when the original image has other formats. For example, original image data having another format may be converted into the RGB format, and the individual color components of the converted data may undergo the simulate illustration process of this embodiment or a filter corresponding to the data format of the original image may be applied. For example, when an original image has a YMC format, the signs of the coefficients of the filters used in this embodiment may be inverted.

Even when an original image has a YHS format or the like,  
the present invention can be applied. For example, when  
an original image has a format such as an Lab format or  
the like that allows extraction of only a luminance  
5 component, only the luminance component can be filtered,  
thus reducing the computation volume compared to that of  
filtering in units of components of the RGB format.

The image to be subjected to simulate illustration  
process in this embodiment is not limited to an image  
10 sensed by the digital camera. For example, a photo image  
scanned by, e.g., a photoscanner or the like may be  
input, or a photo image stored in a medium of an  
external device such as a CD-ROM may be used.

In this embodiment, the filter coefficients are  
15 controlled to implement the simulate illustration  
process. However, an illustration-edited image obtained  
by extracting image edges and by decreasing the number  
of tone levels and by increasing its brightness can also  
be obtained by a method other than filter coefficient  
20 control. For example, a program may be set in advance to  
automatically execute a process for increasing the  
brightness of the entire image and decreasing the number  
of tone levels, after an original image undergoes a  
conversion process such as an oil paint process or the  
25 like to obtain a hand-painted like image by an  
application such as Photoshop (tradename) available from

Adobe Corp., thus obtaining a hand-illustrated like image shown in Fig. 20C.

<Second Embodiment>

The second embodiment of the present invention  
5 will be described below.

In the first embodiment, the simulate illustration process is automatically executed for an original image. In the second embodiment, an image processing system that allows the user to input an arbitrary edit  
10 instruction will be explained.

Since the system arrangement in the second embodiment is the same as that in the first embodiment, a detailed description thereof will be omitted.

Fig. 9 is a flow chart showing the image edit  
15 process in the second embodiment. In Fig. 9, the same step numbers denote the same processes as those in Fig. 3 in the first embodiment. Note that of the processes shown in Fig. 9, processes which require a user interface are implemented by the predetermined  
20 application software programs 101, and other processes are implemented by the image correction processing unit 120 in the printer driver 103.

•Target Image Selection

In step S31, the user selects an image to be  
25 edited. Fig. 10 shows an example of the selection window for an image to be edited, which is displayed on the

monitor 106 upon launching a predetermined image application 101. Referring to Fig. 10, reference numeral 11 denotes an image display frame for displaying a list of reduced-scale images (thumbnail images) of candidate images which may undergo the edit process of this embodiment. These images may be input from the HD 107 or a medium of an external storage device such as a CD-ROM or the like, or may be read out from the RAM 109 or ROM 110. The images displayed on the image display frame 11 are not limited to thumbnail images. For example, an arbitrary one of thumbnail images displayed in a list may be selected, and may be displayed after it is decoded. Reference numeral 14 denotes a button for instructing to start an image edit process; and 15, a button for instructing to end the image edit process. Other function buttons may be provided. Note that text display "list of images" displayed above the "edit" button 14 indicates that the current display on the image display frame 11 is list display of images.

On the target image selection window shown in Fig. 10, the user moves a cursor 16 on the screen onto a desired image on the image display frame 11 by the instruction input device such as a keyboard, mouse, or the like, and makes execution operation such as clicking or the like, thus selecting that image. In Fig. 10, the image displayed on the right end of the image display

frame is selected, and is highlighted by a bold frame.  
Note that a plurality of images may be selected. When  
the user makes execution operation by moving the cursor  
16 onto the "edit" button 14 while at least one image is  
5 selected, the selected image is determined, and the  
start instruction of the image edit process is issued.  
The selected image data is loaded onto a predetermined  
area on the RAM 109 in the RGB format.

•Edit Mode Selection

10 In step S32, the user then designates an edit mode  
(illustration mode in this embodiment) to be executed.  
In this embodiment, the user designates an arbitrary one  
of a plurality of edit modes prepared in advance. An  
edit mode selection process will be explained below with  
15 reference to Fig. 11.

Fig. 11 shows the edit mode selection window,  
which is displayed on the monitor 106 after the image to  
be edited is selected, as described above. Referring to  
Fig. 11, reference numeral 21 denotes an edit mode  
20 display frame for displaying a list of edit modes, which  
can be executed in this embodiment, together with images  
that represent their outlines. The edit mode display  
frame 21 can display the previous or next page upon  
operation of a button 22 or 23, and can display other  
25 edit modes. Reference numeral 27 denotes a button  
pressed upon setting details. Reference numeral 24

denotes a button for instructing to start the image edit process; and 25, a button used to return to the target image selection window shown in Fig. 10. In Fig. 11 as well, other function buttons may be provided. Note that  
5 text display "list of edit modes" displayed above the "return" button 25 indicates that the current display on the edit mode display frame 21 is list display of edit modes.

A description of the detailed contents of the  
10 individual edit modes will be omitted. For example, for edit modes (brick mode and the like) for printing predetermined texture data to overlap an original image, RGB data indicating a predetermined color suitable for the mode is stored on a predetermined area on the RAM  
15 109 or ROM 110 as texture data together with a predetermined pattern.

Note that the images that represent outlines of the individual edit modes displayed on the edit mode display frame 21 in Fig. 11 may be prepared in advance  
20 in the apparatus. For example, the image data to be edited selected in Fig. 10 or its thumbnail data may actually undergo edit processes corresponding to the individual modes, and the process results may be displayed (previewed) on the edit mode display frame 21.

25 On the edit mode selection window shown in Fig. 11, the user moves a cursor 26 on the screen onto a desired

edit mode on the edit mode display frame 21 by the instruction input device such as a keyboard, mouse, or the like, and makes execution operation such as clicking or the like, thus selecting that edit mode. In Fig. 11,  
5 an "illustration" mode is selected and is highlighted by a bold frame.

•Detail setup for Simulate illustration process

When the user further clicks the detail setup button 27 in this state, he or she can arbitrarily set  
10 detailed parameters upon an edit process in the illustration mode, i.e., in the simulate illustration process. Fig. 12 shows an example of the detail setup window. Referring to Fig. 12, reference numeral 31 denotes an image display frame for displaying items that  
15 the user can set. Upon operation of a button 32 or 33, the image display frame 31 can display other items that the user can set. Reference numeral 35 denotes a button used to return to the edit mode selection window shown in Fig. 11. Note that text display "illustration mode  
20 details" indicates that the current display on the image display frame 31 is to set up detailed items in the simulate illustration process.

Fields 38 and 39 displayed within the image display frame 31 are respectively used to brightness  
25 setting and grainy setting parameters. In the brightness setting field 38, the user selects a "high brightness



setting" or "low brightness setting" button depending on whether he or she wants to make the illustration-edited image relatively bright or dark. Of course, a "normal setting" button is selected as a default. A filter is  
5 selected in correspondence with the selected brightness parameter. In the grainy setting field 39, whether or not a grainy edit process is executed is set by either an ON or OFF button. For example, the ON button is selected as a default.

10           Note that setup items shown in Fig. 12 are merely examples, and setup items based on other parameters may be provided. For example, parameters that pertain to setups of filter sizes described in the first embodiment may be set by the user.

15           Upon completion of setups of various detailed items by the user, the user presses the "return" button 35 to display the edit mode selection window shown in Fig. 11 again. Upon clicking the "edit" button 24, the edit mode setup process in step S32 ends, and the  
20 simulate illustration process, which has been selected and set up in detail by the user, is started.

          The grainy edit setups in the second embodiment will be explained below. In the first embodiment described above, an image sensed by the digital camera  
25 111 mainly undergoes simulate illustration process. Hence, since the image to be edited has already

undergone block encoding, e.g., JPEG encoding, block distortion unique to block encoding is produced although it is not so conspicuous. The original image example shown in Fig. 20A in the first embodiment is an image which is JPEG-encoded and then decoded. Hence, in the illustration-edited image shown in Fig. 20C, since block distortion is emphasized by filtering, fine noise components are produced and form grains. However, since grains form, the illustration-edited image becomes a unique one that does not give any monotonous impression. In the second embodiment, such granularity is used as one (grainy effect) of illustration effects, and the user can select whether or not the effect (grainy effect) is added.

15           In step S33 in Fig. 9, the image to be edited selected in step S31 is read out on the basis of the edit mode and detailed parameters set in step S32, and is stored on a predetermined area in the RAM 109 as original image data to be edited.

20   •Original Image Data Generation Process

          A process executed when the illustration mode is selected will be explained below as an example of an original image data generation process in step S33. Fig. 13 is a flow chart showing the original image data generation process in the illustration mode.

It is checked in step S41 if the grainy edit setup parameter is "ON". If the grainy edit setup parameter is "ON", i.e., the user requests to add the grainy effect, the flow advances to step S42 to check if the selected  
5 image to be edited is an image encoded by block encoding. If the image to be edited is a block-encoded image, that image is decoded in step S44. On the other hand, if the image to be edited is not a block-encoded image, the image is encoded by JPEG in step S43.

10 In step S43, the image data to be edited is converted into JPEG codes irrespective of its data format. For example, when an image has been encoded by another format, it is decoded and is then encoded by JPEG to give block distortion. After that, the encoded  
15 image is decoded in step S44.

On the other hand, if it is determined in step S41 that the grainy edit setup parameter is "OFF", since the user requested to turn off the grainy effect, the flow advances to step S45 to check if the selected image to  
20 be edited is an image encoded by JPEG. If the image to be edited is a JPEG-encoded image, that image is decoded in step S46. Furthermore, the decoded image is smoothed in step S47 to remove block distortion. On the other hand, if it is determined in step S45 that the image to  
25 be edited is not a JPEG-encoded image, the image

undergoes, e.g., a decoding process corresponding to its format, thus ending the process.

As can be seen from the flow chart shown in Fig. 13, when the grainy edit setup parameter is "ON",  
5 the processing speed for a JPEG image becomes higher since smoothing in step S47 is skipped.

•Image Edit (Simulate illustration) Process

In step S34, the image edit process in the edit mode set in step S32 is executed for the original image  
10 data.

A simulate illustration process as an example of the image edit process in step S34 will be explained below. Fig. 14 is a flow chart showing the simulate illustration process. The same step numbers in Fig. 14  
15 denote the same processes as those in Fig. 8 described in the first embodiment. Referring to Fig. 14, a filter is set in step S82, S83, or S84 in correspondence with the contents of the brightness setup parameter in the illustration mode, which are set as described above, and  
20 are discriminated in step S81. More specifically, if a "darker edit" setup parameter is selected, the filter 61 as a relatively dark filter is set in step S82; if a "normal edit" setup parameter is selected, the filter 40 as a normal filter is set in step S83; or if a "brighter  
25 edit" setup parameter is selected, the filter 62 as a relatively bright filter is set in step S84. Based on

the selected filter, the original image is filtered in step S85, thus implementing simulate illustration process that the user wanted.

As described above, according to the second embodiment, the user can select whether the grainy edit setup parameter is ON or OFF. Figs. 21A to 21C show examples of the illustration edit result when the grainy edit setup parameter is OFF. Fig. 21A shows an image obtained by decompressing an image which has been compressed in the TIFF format. That is, this image does not have any block distortion unique to JPEG. Fig. 21A shows a standard image (SHIPP) of The Institute of Image Electronics Engineers of Japan, which has  $4,096 \times 3,027$  (around 12.5 million) pixels. Figs. 21B and 21C respectively show the illustration edit results of the image shown in Fig. 21A using  $3 \times 3$  and  $5 \times 5$  filters. For example, as can be seen from Fig. 21C, a monochrome illustration edit result in which production of granularity is suppressed is obtained compared to the grainy image shown in Fig. 20C. As in Figs. 20A to 20C, the image filtered by the  $5 \times 5$  filter shown in Fig. 21C presents the higher illustration edit effect than the image filtered using the  $3 \times 3$  filter shown in Fig. 21B.

As described above, according to the second embodiment, since the user can select an arbitrary edit mode and can setup the details of the selected mode,

simulate illustration process can be achieved in the mode that the user wanted especially in the process. Furthermore, since the histogram generation process and the like in the first embodiment need not be executed,  
5 high-speed processing can be realized.

Note that the second embodiment has exemplified the edit process especially in the illustration mode of a plurality of selectable edit modes. However, the present invention is not limited to such specific edit  
10 mode, and when a plurality of edit processes including simulate illustration process are executed in a predetermined order, a more effective edit result can be obtained.

In the above embodiments, the image edit function  
15 is provided to the printer driver. Of course, this function can be implemented when the OS executes an application.

Also, in the above description, various kinds of information and areas required in the simulate  
20 illustration process are assured on a predetermined area on the RAM 19. The areas may be assured upon launching an application that implements the edit process, or may be assured when they are needed. According to the latter method, since an excessive memory area need not be  
25 assured, a larger memory area can be assigned to other processes, thus improving the memory efficiency.

<Third Embodiment>

The third embodiment of the present invention will be described below.

According to the first and second embodiments, a  
5 simple simulate illustration process is implemented for a photo image by filtering.

However, when this simulate illustration process is implemented by a printer driver, the following problem is posed.

10 More specifically, it is a common practice for serial printers represented by ink-jet printers and the like to execute a raster process to save the memory size used. In this case, since an image is segmented and processed, one processing unit does not always have data  
15 for a plurality of lines, and data for only one line may be processed in an extreme case. Hence, when an image undergoes simulate illustration process by means of filtering, the number of pixels to be filtered other than the pixel of interest often becomes very small.  
20 This means deterioration of the quality of an image obtained as an edit result.

To solve such problem, the third embodiment will exemplify an effective simulate illustration process even for rasterized image data.

25 Since the system arrangement and an outline of processes by the printer driver in the third embodiment

are the same as those shown in Figs. 1 to 3 in the first embodiment, a detailed description thereof will be omitted.

•Simulate illustration Process

5           In the third embodiment as well, by filtering R, G, and B planes of an original image using the filter described in the first embodiment, the original image can be easily converted into a hand-illustrated like image. In the third embodiment, in order to reduce the  
10   computation volume, RGB data are decomposed into luminance and color difference signals, only the luminance signal is filtered, and the color difference signals are multiplied by a given constant in correspondence with desired brightness.

15           Luminance (Y)/chromaticity (C1, C2) conversion are done by:

$$Y = 0.30R + 0.59G + 0.11B)$$

$$C1 = R - Y$$

$$C2 = B - Y$$

20   Note that the luminance and chromaticity conversion equations are not limited to the above ones, but may use various other equations.

          In the third embodiment, the luminance signal Y undergoes, e.g., a simulate illustration process using  
25   the filter 40 shown in Fig. 5. Note that the coefficient (weight) of the pixel of interest can be set at "26" if



the original image has sufficient exposure (standard brightness), i.e., need only comply with a rule that the sum total of coefficients that define the filter is "2". On the other hand, if a relatively dark image is to be  
5 obtained, the coefficient of the pixel of interest can be set at "25"; if a relatively bright image is to be obtained, the coefficient (weight) of the pixel of interest can be set at "27", "28",.... When exposure is inappropriate, the original image may be corrected by  
10 the image correction processing unit 120.

In this case, the user selects a desired brightness level using a user interface shown in, e.g., Fig. 19B, and the coefficient of the pixel of interest is set in accordance with the selected brightness level.  
15 For example, when the user wants to set a standard brightness level, the luminance signal Y is filtered using the filter 40 to obtain Y', and the color difference signals C1 and C2 are edited by calculating:

$$C1' = K \times C1$$

$$20 \quad C2' = K \times C2$$

where K is a constant corresponding to the brightness level designated by the user. For example, when the standard brightness level is set,  $K = 2$ ; when a brighter mode is selected,  $K = 3, 4, \dots$  may be set. K is not  
25 limited to an integer.

The obtained Y', C1', and C2' are re-converted into R, G, and B values, thus completing the simulate illustration process of the third embodiment.

•Measure Against Rasterized Image

5           However, the aforementioned problem is posed. That is, since the serial printer performs so-called rasterization unlike application software that can batch-process a single image, an image is segmented and processed and, hence, a  $5 \times 5$  filter process may be  
10 disabled.

Even when an image can be batch-processed, the upper, lower, right, and left end regions of an image cannot undergo the  $5 \times 5$  filter process. A case will be exemplified below with reference to Fig. 15 wherein the  
15  $5 \times 5$  filter process is disabled. Referring to Fig. 15, reference numeral 700 denotes an image to be processed, in which one rectangle indicates one pixel. Assuming that a pixel 701 of interest located in the upper left region of the image 700 is to be processed, it cannot  
20 undergo, e.g., a  $5 \times 5$  filter process. Therefore, the image quality of this region becomes different from that of other regions that can undergo the  $5 \times 5$  filter process. Such problem occurs in any filter processes. However, since this problem normally occurs only in end  
25 portions of an image, it is not visually conspicuous, and is not so serious.

When such pixel 701 of interest is to be processed,  
a method of adjusting the coefficient of the pixel 701  
of interest in accordance with the number of pixels (to  
be referred to as effective pixels hereinafter) which  
5 are located with a region 702 near the pixel of interest  
and allow  $5 \times 5$  filtering is effective. More  
specifically, as a special filter for the pixel 701 of  
interest, if an output with a standard brightness level  
is required, a filter having a sum total of coefficients  
10 = "2" can be generated complying with a rule that the  
sum total of filter coefficients is positive, and can be  
used in filtering for the region 702. Reference numeral  
703 denotes an example of such special filter.

In any case, when an image is batch-processed, a  
15 filter process can be done based on three or more lines  
or columns of pixels. In addition, since a special  
filter process is done for only image end portions, the  
filter process, i.e., the simulate illustration process  
can be completed hardly forming any visually unnatural  
20 regions. It is empirically known that hardly any  
visually unnatural region is formed as a result of a  
filter process for three or more lines or columns of  
effective pixels.

On the other hand, when image data is rasterized  
25 like in the printer driver, image data for a plurality  
of lines (e.g., three lines or more) are not always

assured as a segmented processing unit. Also, since a single image is processed in units of a plurality of regions, if different filters are used every processing units, image quality differences become conspicuous at  
5 the boundary lines of regions.

In an extreme case, for example, when an image is obliquely set, as shown in Fig. 16A, the image is segmented into regions, as shown in Fig. 16B. In this case, a region 801 that can assure a rectangular region  
10 can undergo a normal filter process for a plurality of lines, but a non-rectangular region 802 must undergo a filter process in units of lines in the worst case. Consequently, processing result differences become noticeable in units of regions. Such problem is  
15 similarly posed when an image is deformed into a special shape, e.g., a heart shape.

Since the principle of the simulate illustration process in the present invention is to emphasize the difference between the pixel of interest and its  
20 neighboring pixels, a portion having an abrupt change between pixels, i.e., an edge portion is emphasized. That is, since the essence of the present invention is to more emphasize the difference the pixel of interest and its surrounding pixels, the aforementioned  
25 shortcomings can be eliminated by adjusting the

coefficients in correspondence with the filter size that can be formed.

Fig. 17 is a flow chart showing the simulate illustration process for rasterized image data in the third embodiment.

Image data for one or a plurality of lines, which has been segmented into a given processing unit by rasterization, is input (S131). A filter is set in correspondence with the number of lines of that image data as the processing unit.

It is checked if the number of lines that form the data to be processed allows a normal filter process (S132). If the normal filter process cannot be done, the flow advances to step S134 to set a special filter. For example, when the data to be processed consists of one line (or one column), a special filter shown in Fig. 18A is used; when the data to be processed consists of two lines (or two columns), a special filter shown in Fig. 18B is used. A coefficient P of the pixel of interest in each of the special filters shown in Figs. 18A and 18B can be set in correspondence with the brightness of an image that the user desires as an edit result. That is, when standard brightness is required, the filter is set so that the sum total of the coefficients is "2". Even when the image data to be

processed consists of one column of pixels in place of one row of them, such filter can be used.

On the other hand, if the number of lines that form the image data to be processed allows the normal  
5 filter process, a normal filter ( $5 \times 5$  filter in this embodiment) is set in step S133.

In step S135, the data to be processed is filtered on the basis of the filter set in step S133 or S134, thus implementing simulate illustration process in the  
10 third embodiment. The processes in steps S131 to S135 are repeated until it is determined in step S136 that the simulate illustration process is complete for all the lines of image data.

Note that the special filters in the third  
15 embodiment are not limited to the examples shown in Figs. 18A and 18B, but may have any other  $p \times q$  sizes ( $p$  and  $q$  are integers equal to or larger than 1).

As can be seen from Figs. 18A and 18B, surrounding pixels other than the pixel of interest are set to have  
20 larger coefficients with decreasing special filter size. The coefficient of the pixel of interest is determined based on those coefficients of the surrounding pixels.

Note that the coefficients of the surrounding pixel in the special filter are preferably set to be the  
25 powers of 2. In this manner, shift operations are

allowed, thus expecting a light arithmetic load and a short processing time.

Note that the number of "effective pixels" may become short even for image data having a number of  
5 lines that allow the normal filter process. In such case, by similarly applying the aforementioned principle, any problems on the boundary lines of segmented regions can be avoided.

As larger coefficients are set, features (edges)  
10 stand out more accordingly. However, since an optimal degree of emphasis depends on balance with a filter used in the normal process, weights are preferably finally adjusted with reference to the output result. For example, the coefficients are preferably changed in  
15 correspondence with the number of effective pixels and their distances from the pixel of interest.

As described above, according to the third embodiment, when the filter coefficients are adjusted in correspondence with the number of lines that can be  
20 processed at the same time even for image data to be rasterized, a natural, continuous image processing result at boundaries of processing regions can be obtained. Hence, even when an image is placed obliquely, or is deformed into an arbitrary shape, it can undergo  
25 appropriate simulate illustration process.

The third embodiment has explained the method of converting RGB information of an image into luminance/color difference information, filtering only the luminance information, and multiplying color  
5 difference components by a constant on the basis of the required brightness. Of course, RGB information itself or image data in other formats may be filtered.

For example, original image data having a format other than the RGB format may be tentatively converted  
10 into the RGB format, and the individual color components may undergo the simulate illustration process of this embodiment, or the original image data itself may undergo a filter process according to its data format. For example, when original image data has the YMC format,  
15 the signs of all filter coefficients used in the third embodiment may be inverted.

Furthermore, the third embodiment can be applied to original image data having the YHS format,  $L^*a^*b^*$  format, and the like. That is, in such formats, since a  
20 luminance component can be easily extracted from an original image, only the luminance component can undergo the filter process, thus reducing the computation volume compared to filter processes in units of components of the RGB format. Especially, this effect becomes higher  
25 with increasing filter size.

<Fourth Embodiment>



The fourth embodiment of the present invention will be described below. Since the system arrangement in the fourth embodiment is the same as that in the first embodiment, a detailed description thereof will be  
5 omitted.

Contrary to the need for higher-definition color image processes that pursue reality of photo images, the need for a monochrome edit process represented by a sepia tone process is still high. The monochrome edit  
10 process itself has prevailed. When an image that has undergone simulate illustration process by the filter process described in the first to third embodiments further undergoes monochrome edit, a unique image like a sketch by means of monochrome crayon can be provided.

15 In the fourth embodiment, the user can set an arbitrary hue upon designating the monochrome edit, and when the user also designates simulate illustration process, the simulate illustration process is done prior to the monochrome edit.

20 For example, the user sets desired image processes using a user interface (to be abbreviated as a UI hereinafter) shown in Figs. 19A to 19C, and can preview the processing result.

Fig. 19A shows a default window of the image  
25 process setup UI. After the user checks a check box of "Simulate illustration", as shown in Fig. 19B, he or she

can adjust a "Contrast" adjustment lever for setting contrast (brightness). In accordance with the adjusted position of this lever, a preview image that has undergone simulate illustration process is displayed.

5            Fig. 19C shows an UI example set with "Monochrome Effects", i.e., the monochrome edit, after the "Simulate illustration" process shown in Fig. 19B. In the fourth embodiment, upon setting the monochrome edit, the user can select a hue such as sepia or the like which is  
10           popularly used, and can also designate an arbitrary hue other than popular ones from the entire hue range using a color bar. Of course, by selecting only "Monochrome Effects", the monochrome edit using achromatic color can be done. According to Fig. 19C, yellowish green is set  
15           as user designated color, and a preview image after the monochrome edit using that designated color is displayed.

            When the user is satisfied with the preview image on the UI, he or she selects an "OK" button to close the detail setup window of the image process, and the setup  
20           contents are held in a RAM (not shown) or the like in a printer driver. In this manner, upon executing the image process, an image that has undergone the edit process the user desires can be obtained on the basis of the held setup contents.

25           As has been explained in the third embodiment, when the simulate illustration process is done in units

of luminance/color difference signals, the color  
difference signals are set in an arbitrary hue, thus  
allowing easy and quick monochrome edit. For example,  
when an original image shown in Fig. 20A undergoes the  
5 simulate illustration process and the monochrome edit  
process using sepia, a unique image shown in Fig. 20D  
can be obtained.

As described above, according to the fourth  
embodiment, since photo image data can undergo the  
10 monochrome edit process using an arbitrary hue in  
addition to the simulate illustration process, even the  
user who has poor knowledge about image processes can  
easily obtain an image that has undergone desired  
processes including many patterns, unique image process  
15 effects, and the like.

Hence, a photo image can undergo an edit process  
in higher degree of freedom, and an image with higher  
originality can be easily created.

In the fourth embodiment, a single image undergoes  
20 the simulate illustration and monochrome edit processes  
as setups of the image processes. Of course, other image  
processes may be simultaneously executed. For example,  
by setting the roughness of the edited image using the  
UI, arbitrary resolution conversion can be achieved.  
25 <Other Embodiments>

Note that the present invention may be applied to

either a system constituted by a plurality of devices  
(e.g., a host computer, an interface device, a reader, a  
printer, and the like), or an apparatus consisting of a  
single equipment (e.g., a copying machine, a facsimile  
5 apparatus, or the like).

The objects of the present invention are also  
achieved by supplying a storage medium, which records a  
program code of a software program that can implement  
the functions of the above-mentioned embodiments to the  
10 system or apparatus, and reading out and executing the  
program code stored in the storage medium by a computer  
(or a CPU or MPU) of the system or apparatus.

In this case, the program code itself read out  
from the storage medium implements the functions of the  
15 above-mentioned embodiments, and the storage medium  
which stores the program code constitutes the present  
invention.

As the storage medium for supplying the program  
code, for example, a floppy disk, hard disk, optical  
20 disk, magneto-optical disk, CD-ROM, CD-R, magnetic tape,  
nonvolatile memory card, ROM, and the like may be used.

The functions of the above-mentioned embodiments  
may be implemented not only by executing the readout  
program code by the computer but also by some or all of  
25 actual processing operations executed by an OS  
(operating system) running on the computer on the basis

of an instruction of the program code.

The present invention includes a product, e.g., a printout, obtained by the image processing method of the present invention.

5           Furthermore, the present invention also includes a case where, after the program codes read from the storage medium are written in a function expansion card which is inserted into the computer or in a memory provided in a function expansion unit which is connected  
10 to the computer, CPU or the like contained in the function expansion card or unit performs a part or entire process in accordance with designations of the program codes and realizes functions of the above embodiments.

15           In a case where the present invention is applied to the aforesaid storage medium, the storage medium stores program codes corresponding to the flowcharts (Figs.3, 4, 8, 9, 13, 14 and 17) described in the embodiments.

20           In addition to JPEG used in the present invention, various other methods (for example, a method called scale index coding) can be used as block coding.

          As many apparently widely different embodiments of the present invention can be made without departing from  
25 the spirit and scope thereof, it is to be understood that the invention is not limited to the specific

embodiments thereof except as defined in the appended claims.